



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/661,167	09/12/2003	Dmitry M. Smirnov	021756-017900US	7522
51206	7590	04/01/2009		
TOWNSEND AND TOWNSEND AND CREW LLP TWO EMBARCADERO CENTER 8TH FLOOR SAN FRANCISCO, CA 94111-3834			EXAMINER	
			GOFMAN, ALEX N	
			ART UNIT	PAPER NUMBER
			2162	
			MAIL DATE	DELIVERY MODE
			04/01/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/661,167	Applicant(s) SMIRNOV ET AL.
	Examiner ALEX GOFFMAN	Art Unit 2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED. (35 U.S.C. § 133).

Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 04 November 2008.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-42,44-48 and 50-68 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-42,44-48 and 50-68 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 12 September 2003 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____
 5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

Amendment submitted November 4, 2008 has been considered by examiner.

Claims 1-42, 44-48 and 50-68 are pending.

Response to Arguments

1. Applicant's arguments with respect to Claims 1-42, 44-48 and 50-68 have been fully considered but they are not persuasive.

As to argument of **Claims 1, 7, 20, and 65**, the Applicant states that Lee does not disclose "concurrently navigating a transform process definition and data to be transformed or otherwise navigating the definition and data at the same time." Specifically, the Applicant states that Lee does not disclose "any reference to concurrency." The Examiner respectfully disagrees.

Lee [0323] transforms "XML data" into "Output XML." It is done by transforming the data by using templates [0304], "Identity Server 40 completes the request translation by retrieving XML templates and XSL stylesheets." For example, if Lee needs to transform data, a "process definition," which is a template for transforming data, is accessed. The Claim limitation states that there are "*one or more* transform process definitions." If there is only one process definition that exists, then the navigating would have be concurrent with the transformation since there is only one process definition that exists. As such, Lee

discloses "concurrently navigating a transform process definition and data to be transformed."

As to specific agreements of **Claim 65**, the Applicant states that Lee does not disclose ""wherein the means for positioning the definition pointer can be invoked concurrently with the means for positioning the payload pointer." The Examiner respectfully disagrees.

A "payload pointer," according to the specification of the instant application [0049], is "used to point to the data currently being transformed." A definition pointer, according to the specification of the instant application [0049], is "used to point to the appropriate process definition in Active XML 160." The same reasoning is applied as to arguments of Claim 1 above. If there is one process definition, then the traversal of a "process definition" and the data to be transformed is concurrent. Also, the pointing to a specific "process definition" is done once a specific template is identified. As such, Lee discloses "wherein the means for positioning the definition pointer can be invoked concurrently with the means for positioning the payload pointer."

As to arguments of **Claims 14 and 57**, the Application states that Lee does not disclose "using the identification data to select a transform process definition." The Examiner respectfully disagrees.

Lee [0301] states "In order to perform request translation (step 1606, FIG. 35), Identity Server 40 maintains program service 1660 and XML data registry 1670... Each function in the request corresponds to at least one program listed in program service." Lee [0302] further states "Each registration file contains information necessary for structuring the output of a program's result." This means that once a translation request is received, the request contains data which communicates with the data registry. The data registry then uses the appropriate template to transform the incoming data. The data registry thus becomes an intermediate program between receiving data to be transformed and actual data transformation. The features of the Claim do not limit the Claim not to have an intermediary program. Thus, Lee discloses "using the identification data to select a transform process definition."

As to arguments of **Claim 57**, the Applicant states that Lee does not disclose "using the identification information to extract a transform process definition from a plurality of transform process definitions." The Examiner respectfully disagrees.

Similar reasoning as to arguments of **Claim 14** applies. Also, Lee [0301] and Figure 37 shows that the XML data registry has a choice between a plurality of "transform process definitions" based on number 1672, 1674 and 1676 in the Figure. Thus Lee discloses the limitations of Claim 57.

As to arguments of **Claims 32 and 62**, the Applicant states that Lee does not disclose "that a transform definition includes a translation codeset parameter." The Examiner respectfully disagrees.

The Claim states that the "codeset parameter enabl[es] the transform[ation] to include a call to one of a function or a lookup table..." Lee [0302] and Figure 37 describes using a data registry file which calls various transformation definitions. The data registry file is a lookup table since it contains data which is used for determining a specific output structure.

As to having a pointer as in **Claim 62**, the pointer, Lee [0302] states "Each registration file in data registry 1670 contains a pointer to an XML template, an XML schema and XSL stylesheet." Thus, Lee discloses the limitations of Claims 32 and 62.

As to arguments of **Claim 44**, the Application states that Lee does not disclose "invoking a parallel process to process sub-definition of the read transform definition." The Examiner respectfully disagrees.

The specification of the instant application [0050] states "the definition pointer may be moved from the compound transform definition at FIG. 4A line 28 to the compound definition at FIG. 4A line 47, skipping the sub-definitions of the compound transform definition at FIG. 4A line 28... the parallel processing thread is typically configured to transform the skipped sub-definitions." The Claims do

not discuss any skipped sub-definitions. Lee [0293] discusses transforming XML data using templates. As such, Lee transforms all data that needs to be transformed including nested data. Thus, Lee discloses the limitations of Claim 44.

As to arguments of **Claim 59**, the Application states that Lee does not disclose "computer instructions to invoke parallel processes to position the definition pointer and increment the payload pointer such that the positioning the definition pointer and incrementing the payload pointer are enabled to occur concurrently." The Examiner respectfully disagrees.

The specification of the instant application [0052] states "The payload pointer is incremented from a prior position until a data element corresponding to the source_field parameter or source_record parameter is found." This means that the incrementing is performed in order to find the appropriate transform definition. Lee [0293, 0301] describes traversing transform definitions in order to properly transform data. This would be done by concurrently traversing the data to be transformed and the transform definition. Thus, Lee discloses the limitations of Claim 59.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-42, 44-48 and 50-68 are rejected under 35 U.S.C. 102(e) as being anticipated by Lee (US Patent Application Publication 2002/0129024), hereinafter, Lee.

Claim 1: Lee discloses a system comprising:

a. a computing device; a data interface on the computing device configured to receive data to be transformed or to send transformed data [0323].

b. memory configured to store one or more transform process definitions having at least one simple transform definition and at least one compound transform definition [0293, 0305, 0312]. [The “transform definition” is the format into which the incoming data will be formatted. As for simple and compound definitions, see example of transformation of data from [0305] into example in [0312].]

c. an application including computer instructions [0293].
d. a data interpreter configured to exchange data with the data interface
and the application, the data interpreter including a transform engine configured
to

d1. select a transform process definition from the one or more transform
process definitions, the selected transform process definition including a
hierarchical data structure [0305, 0312, 0323].

d2. concurrently navigate the selected transform process definition and the
data to be transformed, navigation within the data to be transformed being
responsive to transform definitions within the selected transform process
definition [0304].

d3. generate output data having a data structure responsive to a data
structure of the selected transform process definition [0308].

Claim 2: Lee discloses the system of Claim 1 above, and further discloses
wherein the transformation engine is further configured to process the at least
one compound transform definition using recursion [0247, 0305, 0312].
[Recursion is performed on the nested (subflow) elements.]

Claim 3: Lee discloses the system of Claim 1 above, and further discloses
wherein the data interpreter is further configured to support a plurality of
applications [0307].

Claim 4: Lee discloses the system of Claim 1 above, and further discloses wherein the application is a database application, accounting application, human resources application, customer management application, inventory application, or an internet application [0307].

Claim 5: Lee discloses the system of Claim 1 above, and further discloses wherein the application and the data interpreter are integrated [0307, 0308].

Claim 6: Lee discloses the system of Claim 1 above, and further discloses the data interpreter further includes a computing device configured to support the transform engine [0118].

Claim 7: Lee discloses a data interpreter configured to transform data to be transformed, the data interpreter comprising:

a. at least one computing device [0118].

b. a transform engine supported by the computing device, the transform engine being configured to

b1. access a transform process definition including a hierarchical data structure of transform definitions, the data structure including a simple transform definition and a compound transform definition [0293, 0305, 0312].

b2. concurrently navigate the transform process definition and the data to be transformed, navigation within the data to be transformed being responsive to the transform definition within the transform process definition [0304].

b3. generate output data having a data structure responsive to the transform process definition [0312].

Claim 8: Lee discloses the apparatus of Claim 7 above and further discloses wherein the data structure of the output data is responsive to a structure of transform process definition [0312].

Claim 9: Lee discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is configured to process the compound transform definition using recursion [0247, 0305, 0312].

Claim 10: Lee discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is configured to generate output data including data elements characterized by the transform process definition and having no contribution from the data to be transformed [0308]. [If the transformation template does not contain an output for a specific element, it will not "contribute" that element to the output.].

Claim 11: Lee discloses the apparatus of Claim 7 above and further discloses wherein the transform process definition is configured such that some data elements in the data to be transformed do not make a contribution to the output data [0308]. [If the transformation template does not contain an output for a specific element, it will not "contribute" that element to the output.].

Claim 12: Lee discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is further configured to navigate the data

to be transformed responsive to the date structure of the transform definitions within the transform process definition [0247, 0305, 0312].

Claim 13: Lee discloses the apparatus of Claim 7 above and further discloses wherein the transform engine is further configured to navigate the data to be transformed responsive to content of the transform definitions [0247, 0305, 0312].

Claim 14: Lee discloses a method of transforming data using an application programming interface, the method comprising:

- a. receiving data to be transformed at the application programming interface [0116, 0304].
- b. parsing identification data within the data to be transformed, the identification data characterizing the data to be transformed [0304].
- c. using the identification data to select a transform process definition from a set of one or more transform process definitions, the selected transform process definition defining a process of translating data elements within the data to be transformed to output data elements [0304, 0305, 0312].
- d. transforming the data to be transformed to output data, using a transformation engine and the selected transform process definition, a data structure of the output data being responsive to a data structure of the transform process definition [0305, 0312].

Claim15: Lee discloses the method of Claim 14 above and further discloses wherein transforming the data to be transformed includes nesting of data records [0305,0312].

Claim 16: Lee discloses the method of Claim 14 above and further discloses wherein transforming the data to be transformed includes calling a transformation process recursively responsive to a compound transform definition in the selected transform process definition [0247, 0305, 0312].

Claim 17: Lee discloses the method of Claim 14 above and further discloses wherein the selected transform process definition is selected based on information within the identification data that identifies a destination of the data to be transformed [0304, 0305,0312].

Claim 18: Lee discloses the method of Claim 14 above and further discloses wherein the transform process definition is selected based on information within the identification data that identifies a format of the output data [0304, 0305,0312].

Claim 19: Lee discloses the method of Claim 14 above and further discloses wherein the transform process definition includes a extensible markup language (XML) [0323].

Claim 20: Lee discloses a method of transforming data using an application programming interface, the method comprising:

- a. receiving data to be transformed at the application programming interface, the data to be transformed including identification data [0304].
- b. using the identification data to select a transform process definition from a set of transform process definitions, the selected transform process definition defining a process of translating data elements within data to be transformed to output data elements [0304, 0305, 0312, 0323].
- c. transforming the data to be transformed to output data by concurrently navigating the data to be transformed and the selected transform process definition, navigation in the data to be transformed being responsive to the transform process definition [0304].

Claim 21: Lee discloses the method of Claim 20 above and further discloses wherein the selected transform process definition is selected based on information within the identification data that identifies a format of the data to be transformed [0304, 0305, 0312].

Claim 22: Lee discloses the method of Claim 20 above and further discloses wherein the selected transform process definition is selected based on information within the identification data that identifies a source of the data to be transformed [0304, 0305, 0312].

Claim 23: Lee discloses the method of Claim 20 above and further discloses wherein the application programming interface is shared by several applications [0118].

Claim 24: Lee discloses the method of Claim 20 above and further discloses wherein the application programming interface is shared by several applications and the selected transform process definition is selected based on an identity of one of the several applications [0304].

Claim 25: Lee discloses the method of Claim 20 above and further discloses wherein a structure of the output data is responsive to a structure of a transform definition included in the selected transform process definition [0304, 0305,0312].

Claim 26: Lee discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes nesting of data records [0305,0312].

Claim 27: Lee discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes filtering of data records [0305,0312].

Claim 28: Lee discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes calling a transformation process recursively responsive to a data structure of a transform definition included in the selected transform process definition [0247, 0305, 0312].

Claim 29: Lee discloses the method of Claim 20 above and further discloses wherein transforming the data to be transformed includes searching the

data to be transformed for a data field specified in a transform definition included in the transform process definition [0308].

Claim 30: Lee discloses the method of Claim 20 above and further discloses wherein the transform definition includes a translation codeset parameter configured to invoke an external reference [0307].

Claim 31: Lee discloses the method of Claim 20 above and further discloses wherein the transform definition includes a translation codeset parameter configured to invoke an external reference, the external reference being configured to perform logic operations using the data to be transformed [0307].

Claim 32: Lee discloses a method of transforming data, the method comprising:

a. positioning a definition pointer to point at one of a plurality of transform definitions within a transform process definition [0321]. [The program identifies entries for a name, Bob Smith. The name may serve as a specific identifier like a pointer. Information related to Bob Smith is found and transformed into the appropriate format.]

b. reading the pointed at transform definition [0321].
c. searching data to be transformed for a data element to be transformed, the search being responsive to the pointed at transform definition [0321].

d. transforming any found data element into output data, responsive to the pointed at transform definition, a data structure of the output data being responsive to a data structure of the transform process definition [0321, 0323].

e. wherein the read transform definition includes a translation codeset parameter enabling the transforming to include a call to one of a function or a lookup table located in the pointed at transform definition (Figure 37, [0302]).

Claim 33: Lee discloses the method of Claim 32 above and further discloses determining a type of the read transform definition and, if the transform definition is not a simple transform definition type, recursively calling the method of claim [0247, 0305, 0312].

Claim 34: Lee discloses the method of Claim 32 above and further discloses including determining if all sub-definitions of a compound transform definition have been processed [0293, 0305, 0312].

Claim 35: Lee discloses the method of Claim 32 above and further discloses wherein the method of transforming data includes nesting of a data element [0305, 0312].

Claim 36: Lee discloses the method of Claim 32 above and further discloses if no data element is found in the step of searching data to be transformed, adding an output data element to the output data responsive to the read transform definition, the data to be transformed having no contribution to the output data element [0308].

Claim 37: Lee discloses the method of Claim 32 above and further discloses wherein the read transform definition includes a value parameter configured to specify a value for inclusion in the output data [0318].

Claim 38: Lee discloses the method of Claim 32 above and further discloses wherein the data element is a compound data element and the read transform definition includes a source record parameter configured to specify the compound data element [0308].

Claim 39: Lee discloses the method of Claim 32 above and further discloses wherein the read transform definition is in a meta-language format [0308, 0312].

Claim 40: Lee discloses the method of Claim 32 above and further discloses wherein the data to be transformed data is in a meta-language data format [0308].

Claim 41: Lee discloses the method of Claim 32 above and further discloses wherein the read transform definition includes a transform element having an output field name and a source field parameter [0305, 0308, 0312].

Claim 42: Lee discloses the method of Claim 32 above and further discloses wherein the read transform definition includes a value parameter configured to populate a field in the output data [0305, 0308, 0312].

Claim 44: Lee discloses a method of transforming data, the method comprising:

- a. positioning a definition pointer to point at a transform definition, the transform definition being one of a plurality of transform definitions within a transform process definition [0321].
- b. reading the pointed at transform definition [0321].
- c. positioning a payload pointer to point at a data element to be transformed, the positioning being responsive to a data structure of the transform process definition [0321].
- d. transforming the data element into output data, responsive to the read transform definition [0321].
- e. invoking a parallel process to process sub-definitions of the read transform definitions [0293].

Claim 45: Lee discloses the method of Claim 44 above and further discloses determining a type of the read transform definition and, if the read transform definition is not a simple transform definition type, recursively calling the method of claim 44 [0247, 0305, 0312].

Claim 46: Lee discloses the method of Claim 44 above and further discloses determining a type of the read transform definition and, if the read transform definition is not a simple transform definition type, recursively calling the method of claim 44, wherein the recursive call is responsive to the data structure of the transform process definition [0305, 0308, 0312].

Claim 47: Lee discloses the method of Claim 44 above and further discloses further including determining a type of the read transform definition, if the read transform definition is not a simple transform definition type recursively calling the method of claim 44, and determining if all sub-elements of a compound element have been transformed [0305, 0308, 0312].

Claim 48: Lee discloses the method of Claim 44 above and further discloses further including determining if all sub-elements of a compound element have been transformed and, if the determination returns a value of YES, returning to a calling process [0305, 0308, 0312].

Claim 50: Lee discloses the method of Claim 44 above and further discloses wherein the method of transforming data includes un-nesting of the data element to be transformed [0305, 0308, 0312].

Claim 51: Lee discloses the method of Claim 44 above and further discloses wherein the read transform definition includes a source field parameter configured to specify the data element [0308].

Claim 52: Lee discloses the method of Claim 44 above and further discloses wherein the read transform definition includes a source record parameter configured to specify the compound data element [0305, 0308, 0312].

Claim 53: Lee discloses the method of Claim 44 above and further discloses wherein the read transform definition includes a translation codeset

configured for calling computer instructions including logic operations [0305, 0308, 0312].

Claim 54: Lee discloses the method of Claim 53 above and further discloses wherein the computer instructions are configured to call an external process [0307].

Claim 55: Lee discloses the method of Claim 53 above and further discloses including a step of combining the data element with the transform process definition prior to transforming the data element to output data [0308].

Claim 56: Lee discloses the method of Claim 53 above and further discloses wherein the transform process definition includes a tree data structure [0139].

Claim 57: Lee discloses a method of preparing data for transformation, the method comprising:

- a. receiving data to be transformed [0307].
- b. parsing the received data to determine identification information [0307].
- c. using the identification information to extract a transform process definition from a plurality of transform process definitions, the extracted transform process definition including a transform definition configured to transform the data to be transformed, to direct navigation within the data to be transformed during transformation, and to determine a data structure of output data resulting from transformation of the data to be transformed, the transform definition

including a hierarchical data structure having at least one simple transform definition and at least one compound transform definition, the compound transform definition being configured to generate a compound data element in the output data [0305, 0307, 0312].

d. adding the extracted transform process definition to meta-language transform input data including the data to be transformed [0305, 0307, 0312].

Claim 58: Lee discloses the method of Claim 59 above and further discloses wherein the extracted transform process definition is in a meta-language format [0305, 0307, 0312].

Claim 59: Lee discloses a computer readable media having embodied thereon data, the data comprising:

a. computer instructions configured to position a definition pointer to point at a transform definition, the transform definition being one of a plurality of transform definitions within a transform process definition [0305, 0307, 0312].

b. computer instructions configured to read the pointed at transform definition [0305, 0307, 0312].

c. computer instructions configured to increment a payload pointer, within the data to be transformed, to a data element to be transformed, the incrimination being responsive to the pointed at transform definition [0305, 0307, 0312].

d. computer instructions configured to transform any found data element into output data, responsive to the pointed at transform definition, a data

structure of the output data being responsive to a data structure of the transform process definition [0305, 0307, 0312].

e. computer instructions to invoke parallel processes to position the definition pointer and increment the payload pointer such that the positioning the definition pointer and incrementing the payload pointer are enabled to occur concurrently [0293, 0301].

Claim 60: Lee discloses the media of Claim 59 above, and further discloses wherein the data further comprises computer instructions configured to employ recursion to transform a compound data element within the data to be transformed [0305, 0307, 0312].

Claim 61: Lee discloses the media of Claim 59 above, and further discloses wherein the data further comprises computer instructions configured to transform the data to be transformed using parallel processes [0305, 0307, 0312].

Claim 62: Lee discloses a computer readable storage media having embodied thereon data, the data comprising:

a. payload data including data to be transformed, the data to be transformed including metadata characterizing simple data elements and compound data elements [0305, 0307, 0312].

b. a transform process definition including a transform definition configured to transform the data to be transformed, to direct navigation within the data to be

transformed during transformation, and to determine a data structure of output data resulting from the transformation, the transform definition including a hierarchical data structure having at least one simple transform definition and at least one compound transform definition, the transform definition also including a pointer to a function enabled to perform logical operations on the data to be transformed during transformation and generate the output data, the compound transform definition being configured to generate a compound data element in the output data [0305, 0307, 0312].

Claim 63: Lee discloses the media of Claim 62 above, and further discloses wherein the computer readable media includes memory included in a data interface [0118].

Claim 64: Lee discloses the media of Claim 62 above, and further discloses wherein the computer readable media includes a hard drive [0118].

Claim 65: Lee discloses an application system comprising:

- means for positioning a definition pointer to point at a transform definition within a transform process definition [0321].
- means for reading the transform definition [0321].
- means for positioning a payload pointer to point to a first data element, the first data element being a member of a plurality of data elements within data to be transformed [0305, 0307, 0312].

d. means for generating output data using the first data element and the transform definition [0305, 0307, 0312].

e. wherein the means for positioning the definition pointer can be invoked concurrently with the means for positioning the payload pointer [0304].

Claim 66: Lee discloses the system of Claim 65 above, and further discloses means for selecting the transform process definition from a set of transform process definitions, responsive to data associated with the data to be transformed [0308].

Claim 67: Lee discloses the system of Claim 65 above, and further discloses wherein a second data element has no contribution to output data generated using the transform process definition, the second data element being a member of the plurality of data elements [0308].

Claim 68: Lee discloses the system of Claim 65 above, and further discloses means for adding data to the output data, the added data being configured responsive to the transform process definition and having no contribution from the data to be transformed [0308].

Conclusion

4. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ALEX GOFFMAN whose telephone number is (571)270-1072. The examiner can normally be reached on Mon-Fri 9am-3pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (571)272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Alex Gofman
Examiner
Art Unit 2162

AG
3-23-09

/John Breene/

Supervisory Patent Examiner, Art Unit 2162